

Amélioration des Délais dans les Réseaux à Débits Garantis pour des Flux Temps-Réel Sous Contrainte « (m,k)-Firm »

Koubâa Anis, Yé-Qiong Song

LORIA – UHP Nancy I - INPL - INRIA Lorraine

2, av. de la Forêt de Haye
54516 Vandoeuvre – France

akoubaa@loria.fr

song@loria.fr

Résumé :

L'ordonnanceur Weighted Fair Queueing (WFQ) est déployé dans les réseaux à intégration de services grâce à ses propriétés de garantie de bande passante et de délai borné pour des applications temps-réel. Cependant, la borne sur le temps de réponse fournie à une session est étroitement liée au coefficient de partage de la bande passante. Pour avoir un délai d'attente court, un flux doit réserver une large bande passante. Pour un flux de faible débit moyen et ayant une grande rafale ceci peut conduire à une mauvaise utilisation de la bande passante. Dans ce papier, nous proposons un nouvel algorithme d'ordonnancement appelé (m,k)-WFQ qui intègre les contraintes temporelles (m,k)-firm dans l'algorithme de WFQ standard afin d'améliorer la qualité de service temporelle des flux temps réel ayant de tel contraintes. Une analyse par simulation est réalisée montrant l'amélioration des performances de l'algorithme proposé par rapport à celles du WFQ standard.

MOTS-CLES : (m,k)-firm, WFQ, Simulation, service garanti

1. Introduction

Un réseau d'aujourd'hui doit supporter des applications ayant des contraintes diverses y compris les contraintes du temps réel. Pour fournir une certaine garantie de service (QoS : Quality of Service) vis à vis d'un flux, les algorithmes d'ordonnancement des trafics tel que WFQ (Weighted Fair Queueing) [DKS 89] et ses variantes [WWL 02] ont été développés et certains sont déployés dans les réseaux multimédias pour fournir une bande passante constante à chaque flux servi et assurent une borne maximale sur son

temps de réponse sachant que chaque flux est borné par une courbe d'arrivée maximale (Arrival Curve) [6-7]. Ces techniques de service sont implémentées dans les protocoles de gestion de qualité de service dans Internet tel que *IntServ* et *DiffServ*.

Cependant, le temps de réponse fournie par de tels algorithmes est fonction de la taille de rafale du trafic et de la bande passante réservée. En effet, si un flux arrive en rafale, le délai de bout en bout augmente linéairement avec la taille maximale de rafale. Ceci peut amener au non-respect de l'échéance temporelle des paquets temps-réel. Pour arriver à respecter les échéances il va falloir augmenter la bande passante largement au-dessus du débit moyen du flux. Ce qui résulte en un faible taux d'utilisation de la bande passante globale du réseau. Par exemple, une communication téléphonique via Internet utilisant le protocole *Voix sur IP* est caractérisée par un trafic en rafale ayant un débit moyen assez réduit ($\approx 64 \text{ Kbit/s}$) et pourtant nécessite un délai très restreint pour assurer la bonne interactivité de cette application.

D'autre part, Hamdaoui et Ramanathan dans [HR 95, R 99] ont signalé que les flux temps-réel tels que les flux multimédias peuvent tolérer la perte de quelques paquets si cette perte se fait adéquatement. C'est la classe de flux ayant des garanties (m,k)-firm qui exige qu'au moins m paquets parmi k paquets transmis consécutivement doivent respecter leur échéances temporelles. Cette contrainte définit le modèle de perte de paquets d'une manière précise. Par exemple, dans un flux vidéo temps-réel MPEG, la perte de quelques paquets (trames de fond par exemple) peut être tolérée. Ceci vient du fait que les flux temps-réels sont sensibles aux variations du délai mais tolèrent la perte de quelques paquets.

Plusieurs techniques dans la littérature ont été développées pour ordonnancer des flux ayant des contraintes de type (m,k)-firm. Certes le plus connu est DBP (*Distance Based Priority*) [HR 95] qui est un algorithme dynamique qui sert, à un instant donné, le paquet le plus proche à l'état d'échecs. Aussi, dans [SKPW 03], une technique appelée Matrix DBP est développée pour améliorer le comportement de DBP et réduire le nombre des états d'échecs. Ces algorithmes sont de type Best-Effort et ne permettent pas de fournir un service garanti aux flux ayant des contraintes (m,k)-firm.

Pour améliorer les délais de bout en bout des flux temps-réel arrivant en rafale et ayant des contraintes (m,k)-firm, nous avons développé un nouvel algorithme d'ordonnancement qui combine WFQ avec les paramètres de (m,k)-firm. La solution proposée dans ce papier tire profit de la tolérance de quelques pertes d'échéances en se basant sur les contraintes (m,k)-firm qui définit un moyen pour écarter les paquets selon un motif (m,k)-firm noté μ -pattern. L'idée innovatrice est de classer les paquets d'un flux en paquets optionnels et critiques selon la contrainte (m,k)-firm sur ce flux. Un paquet critique est plus prioritaire qu'un paquet optionnel. Un paquet optionnel n'est donc transmis que lorsqu'il peut respecter son échéance et qu'il n'y a pas la présence des paquets critiques au moment de sa transmission, sinon il sera rejeté. Un paquet critique est transmis dans tous les cas (on doit dimensionner la bande passante selon le sous-flux critique et l'échéance). Notons qu'un paquet optionnel à échéance infinie n'est jamais rejeté ce qui permet de garantir une transmission sans perte d'un flux tel que FTP. La sélection d'un paquet pour le service est basée sur son estampille calculée comme dans le cas de WFQ (*Finish Tag*) mais sa transmission dépend de la classe attribuée à ce paquet (Critique ou Optionnel) et de son échéance.

Cette approche permet aussi de différencier entre le trafic temps-réel et le trafic best-effort en considérant ce dernier comme optionnel.

2. Notions Préliminaires

2.1. Les Contraintes (m,k)-Firm

Un système a des contraintes de type (m,k)-firm, si au moins m paquets doivent respecter leurs échéances dans n'importe quel fenêtre de k paquets consécutifs. Ce système est en état d'échec si et seulement s'il existe plus que (k-m) paquets qui ont raté leurs échéances dans une fenêtre de taille k paquets.

Dans ce papier, nous utilisons les contraintes (m,k)-firm pour définir le concept du μ -pattern comme étant le profil de perte suivant la contrainte définie par le couple (m,k). Il spécifie l'organisation souhaitée des pertes des échéances dans une fenêtre de k paquets.

Définition : μ -pattern

Le μ -pattern d'un flux ayant une contrainte temporelle (m,k)-firm est défini par la succession de k éléments de l'alphabet $\Delta = \{0,1\}$ tel que :

$$\begin{cases} 0 & \text{indique un paquet optionnel} \\ 1 & \text{indique un paquet critique} \end{cases}$$

On représente par $\mu(i)$ le $i^{\text{ème}}$ élément du μ -pattern pour $1 \leq i \leq k$.

En utilisant le concept du μ -pattern, on divise les paquets d'un flux en deux classes : optionnelle et critique. Avec cette classification, il suffit de transmettre avec succès les paquets critiques pour satisfaire cette contrainte. Cependant, il reste toujours la possibilité de transmettre des paquets optionnels sans violer cette contrainte. En général, les paquets optionnels peuvent être transmis ou rejetés suivant les besoins de l'algorithme d'ordonnancement.

De ce fait, le $n^{\text{ième}}$ paquet d'un flux ayant des contraintes temporelles (m,k)-firm, est considéré comme étant un paquet critique si et seulement si il satisfait la relation suivante :

$$\mu(n \% k) = 1 \quad 2.1$$

avec $n \% k$ est le reste de la division de n modulo k .

3. L'Ordonnancement Equitable WFQ

WFQ est un algorithme d'ordonnancement qui dérive de la version fluide GPS. Il consiste à garantir à chaque flux servi la proportion de la bande passante réservée selon son coefficient de partage Φ_i . Dans WFQ le temps virtuel de départ est défini par

$$F_i^k = \max\{F_i^{k-1}, V(t)\} + \frac{L_i^k}{\Phi_i} \quad 3.1$$

avec

F_i^k : temps virtuel de départ du $k^{ième}$ paquet de l' $i^{ième}$ flux,

$V(t)$: le temps virtuel quand le $k^{ième}$ paquet arrive,

Φ_i : le coefficient de partage de l' $i^{ième}$ flux,

L_i^k : la taille du $k^{ième}$ paquet de l' $i^{ième}$ flux,

$\max\{F_i^{k-1}, V(t)\}$: le temps virtuel du début de service du $k^{ième}$ paquet.

La sélection du paquet à servir se fait dans l'ordre croissant des temps virtuels de départ. Il est montré dans [BT 02] que pour un flux S_i ayant un débit moyen réservé r_i et une taille maximale de rafale b_i , le délai garanti par WFQ à ce flux est borné par :

$$D_{i,\max} = \frac{b_i}{r_i} + \frac{L_{\max}}{C} \quad 3.2$$

Où, L_{\max} , est la taille maximale du paquet parmi tous les paquets dans tous les flux et C étant la capacité de traitement du serveur.

4. L'algorithme (m,k)-WFQ

L'algorithme (m,k)-WFQ consiste à intégrer les contraintes temporelle (m,k)-firm au processus d'ordonnancement de WFQ. Cet algorithme est présenté dans la table 1.

L'algorithme de WFQ standard sélectionne les paquets selon l'ordre croissant de leurs temps virtuels de départ. Dans notre approche, l'ordonnanceur doit prendre en compte les contraintes temporelle (m,k). Pour ce faire, les paquets sont initialement marqués comme étant paquets critiques ou paquets optionnels selon le μ -pattern correspondant. La technique de marquage suit l'équation 2.1. De plus, l'ordonnanceur (m,k)-WFQ estampille le paquet par son temps virtuel de départ décrit par l'équation 3.1.

Le processus de service est activé quand au moins un paquet existe dans la file d'attente du système. Le serveur sélectionne le paquet ayant le plus petit temps virtuel de départ parmi tous les **paquets critiques** présents en tête de leur files. Si aucun paquet critique existe, le choix sera fait parmi les paquets optionnels. Après, si le paquet sélectionné est critique, il est transmis directement par le serveur, sinon, si le paquet est optionnel, l'ordonnanceur vérifie avant la transmission si ce paquet pourrait éventuellement satisfaire son échéance. Si l'échéance souhaitée est ratée, le serveur rejette le paquet et refait une nouvelle sélection, sinon, il l'envoi.

L'avantage de l'algorithme proposé est qu'il permet de garantir une bande passante à un flux tout en intégrant les propriétés temporelles dans le processus d'ordonnancement ce qui résulte à gérer les flux plus efficacement. En effet, le rejet des paquets optionnels qui ne satisfont pas leurs échéances permet au serveur de donner la main plus rapidement aux paquets critiques en attente. Cette perte ne dégrade pas les performances des flux servis tant que leurs contraintes (m,k)-firm sont satisfaites. Ainsi, (m,k)-WFQ diminue forcément les bornes sur les temps de réponse des flux temps réels par rapport à WFQ standard.

Table 1. Algorithme (m,k)-WFQ

Entrées
 Flux $S_i = \{(\text{Période ou Débit}), \text{Echéance Désirée}, (m,k), (\text{Gigue ou Rafale}), \text{Taille de Paquet}\}$

Affectation de priorité
 A l'arrivée du $a^{\text{ième}}$ du flux [i] {
 si $(\mu((a\%k_i)+1)=1)$ alors {
 Marquer le paquet comme critique;
 }
 sinon {
 Marquer le paquet comme optionnel;
 }
 Calculer le temps virtuel de départ F_i^k ;
 Estampiller le paquet avec F_i^k ;
 Mettre le paquet dans sa fille d'attente;
 }

Discipline de Service
 Serveur = libre;
 Tant que (la file est non vide) {
 si (serveur!= occupé) {
 Sélectionner le paquet ayant le plus petit temps virtuel de départ
 si (paquet est critique) {
 Envoyer le paquet;
 Serveur = occupé;
 }
 sinon { //Paquet Optionnel
 si (l'échéance serait ratée){
 Rejet du paquet;
 Serveur = libre;
 }
 sinon {
 Envoyer le paquet;
 Serveur = occupé;
 }
 }
 }
 si (serveur== occupé) {
 attendre jusqu'à transmission totale
 du paquet;
 Serveur= libre;
 }
 }

La figure 4.1 montre un exemple d'exécution d'un serveur (m,k)-WFQ. Le symbole 'M' en haut d'un paquet indique que le paquet est considéré comme critique et le symbole 'O' pour les paquets optionnels. Le symbole '1' montre un que le paquet correspondant satisfait son échéance alors que le '0' pour les paquets qui ont raté leurs échéances.

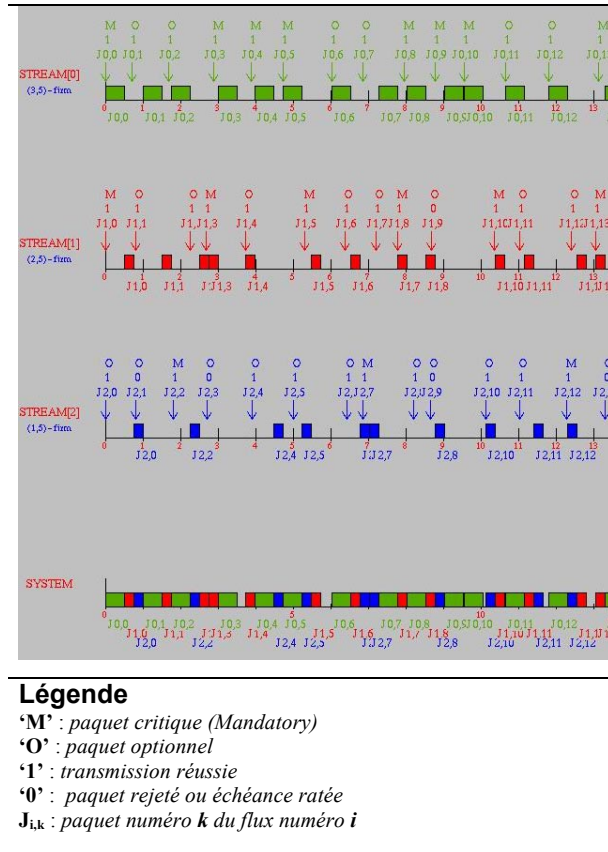


Fig. 4.1. Ordonnanceur (m,k)-WFQ

Dans cet exemple, tous les paquets $J_{i,7}$, $0 \leq i \leq 2$ ont le même temps virtuel de départ $FT_{i,2} = 32$. Dans ce cas, le serveur WFQ standard servira en premier le flux ayant le plus grand coefficient de partage de bande passante ou bien en utilisant une politique de priorité fixe. Cependant, dans ce cas, puisque $J_{0,7}$ et $J_{1,7}$ sont des paquets optionnels et le paquet $J_{2,7}$ est critique alors (m,k)-WFQ transmettra ce paquet avant les autres. Après avoir transmis $J_{2,7}$, (m,k)-WFQ décide s'il va transmettre ou non les deux paquets optionnels en attente. Dans cet exemple, les deux trames optionnelles sont transmises parce qu'elles satisfont leurs échéances respectives. Cependant, les paquets optionnels $J_{1,9}$ et $J_{2,9}$ sont rejetés par l'ordonnanceur pour donner la main plus rapidement aux paquets critiques, déjà en attente, $J_{0,9}$ et $J_{0,10}$.

5. Evaluation de Performance

5.1. Présentation du système

Le système étudié est un ensemble de trois sources qui partagent un lien de 10 Mb/s selon leurs coefficients de réservation : Une source de voix à 64 Kbit/s, une source vidéo à 2Mbit/s et le reste de la capacité du serveur est consommé par un trafic FTP. La table 2 présente la charge générée par chacune des trois sources.

Table 2. Paramètres de Simulation

source	(m,k)	Débit	Trafic	μ - pattern	Echéance Désirée
Voix	(4,5)	64 kb/s	ON/OFF (500/755/50)ms Pseudo	11011	10 ms
Vidéo	(3,5)	2Mb/s	Périodique ~2Mb/s Pseudo	10110	4 ms
FTP	(0,1)	7,936 Mb/s	Périodique ~7.936 Mb/s	0	Inf

Dans cette simulation, on considère une taille fixe à tous les paquets des trois flux de 8 Kbits.

Ce scénario comporte deux sources temps réel (voix et vidéo) et une source n'ayant pas des contraintes temporelles strictes (FTP). Cependant, on configure l'ordonnanceur (m,k)-WFQ de ne rejeter que les paquets optionnels des sources temps-réel, voix et vidéo. En effet, le flux FTP est vulnérable à la perte de paquets. Ainsi, comme tous les paquets de ce flux sont optionnels, le serveur est averti de faire passer tous ses paquets. Ceci est réalisé en mettant une échéance infinie pour les flux best-effort.

Dans la section suivant, nous montrons une étude comparative entre (m,k)-WFQ et WFQ standard afin de montrer l'importance de tenir compte des contraintes temporelles de type (m,k)-firm afin d'améliorer le temps de réponse des applications temps réels sans dégrader celles des autres applications (dans ce cas FTP).

5.2. Résultats de la simulation

Le tableau 3 montre les bornes mesurées sur le temps de réponse des paquets pour chacun des flux fournis par les deux serveurs.

Table3. Bornes sur les délais (ms)

	(m,k)-WFQ	WFQ
Voix	9,769	4776,83
Vidéo	3,999	41,084
FTP	3,837	18,048

On remarque que le flux de la voix souffre d'un délai assez large quand il est servi par WFQ standard. Ceci est prévu du fait que ce flux a un coefficient de partage très petit (proportionnel à $64/10^3$) par rapport aux autres flux d'autant plus sa nature en rafale.

Cependant, en intégrant les contraintes temporelles de type (m,k), l'algorithme (m,k)-WFQ permet d'améliorer le délai maximal d'un paquet du flux de la voix en rejetant les paquets optionnel des flux suivant leurs contraintes (m,k)-firm. Il est alors clair que le rejet des paquets optionnels ne satisfaisant pas leurs échéances, améliore nettement le délai des paquets critiques.

Un autre avantage fourni par l'algorithme (m,k)-WFQ est de différencier entre les flux temps réel et les flux best-effort. Nous avons défini les paquets du flux Best-Effort, FTP, comme étant tous optionnels. Alors, puisque le serveur (m,k)-WFQ sert en premier les paquets critiques, il est certain que les paquets temps-réel critiques sont servis en avance du flux « gourmand » FTP grâce aux propriétés temporelles définies par les contraintes (m,k)-firm. C'est pour cette raison que nous remarquons une nette amélioration sur les délais des flux temps-réel et que tous les paquets qui passent sont servis avant leurs échéances désirées.

L'échéance requise de chaque flux a une influence sur les bornes des délais des flux temps réels. Ceci est montré par le tableau 4 où nous avons changé l'échéance désirée du flux Vidéo à $D_{\text{vidéo}} = 40\text{ms}$ (2^{ème} scénario).

Table 4. Bornes sur les délais (ms)
(2^{ème} Scénario)

	(m,k)-WFQ
Voix	3,177
Vidéo	37,621
FTP	26,111

Ce changement des délais de chacun des flux est dû essentiellement au taux de rejet. En effet, en augmentant l'échéance du flux vidéo, le serveur (m,k)-WFQ servira plus de paquets de ce flux (les paquets optionnels) plus que la condition de rejet devient plus souple quand l'échéance est plus grande. Ce qui explique l'augmentation du délai du flux FTP. La diminution du délai du flux de la voix est mieux expliquée par le tableau 5 qui montre les taux de rejets pour chacun des flux temps-réel dans les deux scénarios.

Table 5. Taux de rejets (ms)

	1 ^{er} Scénario	2 ^{ème} Scénario
Voix	6,8%	15%
Vidéo	5,5%	0%

Nous remarquons que dans le deuxième scénario le taux de rejet des paquets vidéo est nul. C'est à dire, tous les paquets critiques et optionnels de ce flux sont servis, ce qui explique la hausse de la borne sur le délai du trafic FTP, dont tous les paquets sont optionnels. En effet, dans le 1^{er} scénario le rejet des paquets optionnels ayant dépassés leurs échéances du flux vidéo accélère l'envoi des paquets du flux « gourmand » de FTP.

Cependant, la diminution mesurée sur la borne du délai dans 2^{ème} scénario pour le trafic de la voix, s'explique par l'augmentation du taux de rejet des paquets optionnels relatifs à ce flux de 6,8% à 15%. Cette augmentation est légère et la simulation montre que la contrainte temporelle (4,5)-firm est satisfaite dans les deux scénarios, puisque dans les deux cas la borne mesurée sur le délai est inférieure à celle demandée.

La Figure 5.1 montre les temps moyens de réponse de chacun des trois flux expérimentés dans le premier scénario

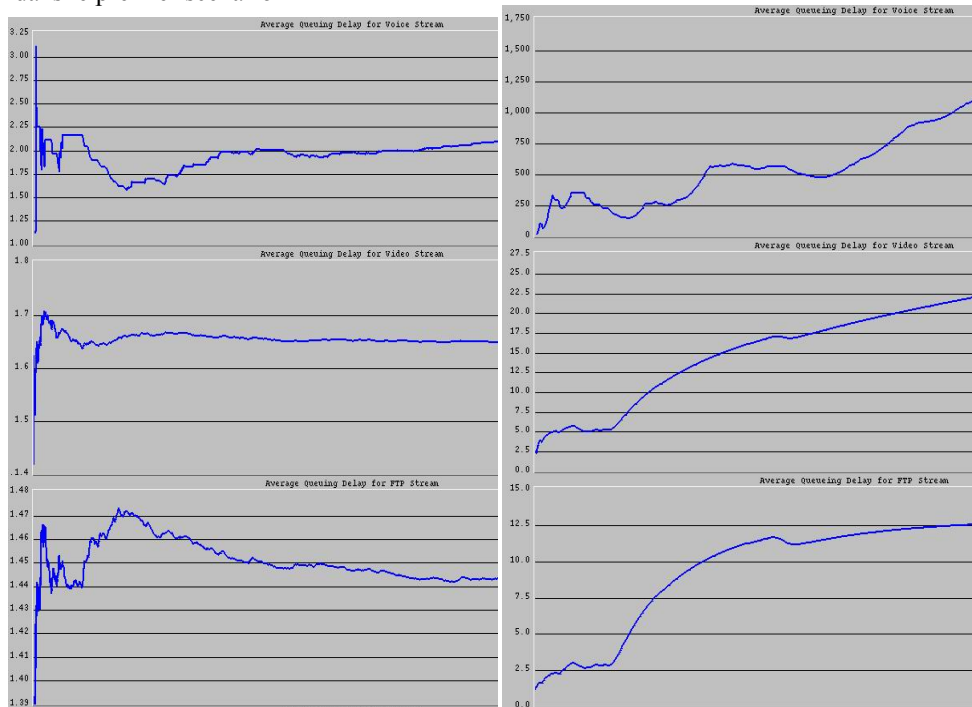


Fig. 5.1. Temps moyen d'attente avec (m,k)-WFQ (gauche) et WFQ (droite)

La nature en rafale du flux de la voix explique bien la non stabilité des temps moyens d'attente dans le système ordonnancé par WFQ. Cet effet négatif est lissé avec l'algorithme de (m,k)-WFQ et les temps moyens de réponse sont stationnaires et plus courts.

WFQ standard pourrait satisfaire un délai plus court au trafic de la voix en lui réservant une bande passante plus grande que son débit moyen. Cependant, cette technique n'est pas efficace car ce gain dégrade les temps de réponse des autres flux temps-réel. L'apport de l'algorithme (m,k)-WFQ est de garantir des bornes plus petites sur les délais de tous les flux servis sans favoriser le service d'un flux sur un autre. L'équité de (m,k)-WFQ est garantie sur le plan de la bande passante et aussi, sur les contraintes temporelles des flux.

6. Conclusion

Nous avons montré dans cette étude menée par simulation l'amélioration du délai des flux temps-réel servis par des serveurs à débits garantis comme WFQ en intégrant les contraintes temporelles (m,k) -firm. Ceci aboutit à la technique d'ordonnancement (m,k) -WFQ.

Les résultats de simulation montrent l'avantage apporté par cette approche en terme de réduction des temps de réponse, aussi bien que la différenciation entre le trafic temps réel et le trafic best-effort. De plus, nous avons noté qu'un même système, initialement instable, peut devenir stable et fournir des délais moyens stationnaires quand les contraintes (m,k) -firm sont prises en compte par le serveur WFQ.

Les travaux en cours consiste à borner analytiquement le délai d'un flux temps réel ayant des contraintes (m,k) -firm et servis par un serveur (m,k) -WFQ. Pour ce fait, nous utilisons le formalisme du Network Calculus qui permet d'étendre les résultats des bornes garanties par WFQ [BT 02] au serveur (m,k) -WFQ.

7. Bibliographie

- [DKS 89] A. Demers, S. Keshav, S. Shenker, "*Analysis and Simulation of Fair Queuing Algorithm*", Proceeding ACM SigComm 89, pp 3-12, 1989
- [WWL 02] S. Wang, Y. Wang, K. Lin, "*Integrating Priority with Share in the Priority-Based Weighted Fair Queueing Scheduler for Real-Time Networks*" Journal of RTS p119-149, 2002
- [HR 95] M. Hamdaoui and P. Ramanathan. "*A dynamic priority assignment technique for streams with (m, k) -firm deadlines*". *IEEE Transactions on Computers*, 44(4), 1443-1451, Dec. 1995.
- [BB 01] G. Bernat and A. Burns. "*Weakly_Hard real-time systems*", *IEEE Transactions on Computers*, 50(4), pp.308-321, 2001.
- [R 99] P. Ramanathan "*Overload Management in Real-Time Control Applications Using (m,k) -Firm Guarantees*" *IEEE Transactions on Parallel and Distributed Systems* Vol 10 No 6, p549-559, June 1999
- [BT 02] JY Le Boudec, P. Thiran, "*Network Calculus : A Theory of Deterministic Queueing Systems for the Internet*" Springer Verlag July 2002 (Version On-Line)
- [KQ 03] Anis Koubaa, Ye Qiong Song "*Upper Bound Evaluation and Improvement of Response Time For Real-Time Communication*" Internal Report LORIA-INRIA 2003
- [SKPW 03] YQ Song, A. Koubaa, E. Poggi, Z. Wang "*Matrix-DBP for On-Line Non-Pre-emptive Scheduling of (m,k) -Firm Constrained Streams*" Real Time System RTS Paris Avril 2003